

**We Claim**

- 5 1. A method (400) of preloading data on a cache (210) in a local machine (235), wherein said cache is operably coupled to a data store (130) in a remote host machine (240), the method characterised by the steps of:
- determining a user behaviour profile for said
- 10 local machine (235);
- predicting (405) a time for data to be required by a user;
- retrieving data relating to said user behaviour profile from said data store (130) in response to a
- 15 predicted time;
- calculating a safety margin of time; and
- preloading said retrieved data to said cache (210), at a time at or before said safety margin prior to said predicted preload time, such that said data is made
- 20 available to a user of said cache when desired.
2. The method (400) of preloading data on a cache (210) according to Claim 1, wherein said step of determining is performed by a preload function (255) in
- 25 said local machine 235 operably coupled to said cache and/or a preload function (265) in a remote host machine (240) operably coupled to said data store (130).
3. The method (400) of preloading data on a cache
- 30 (210) according to Claim 2 further characterised by the step of:

predicting, by at least one preload function, a data type required by said cache user based on said determined user behaviour profile.

5 4. The method (400) of preloading data on a cache (210) according to Claim 1 further characterised in that the step of predicting (405), is performed by said at least one preload function, and comprises predicting (405) an event time for said data type to be required by  
10 said user based on said determined user behaviour profile (210).

5. The method (400) of preloading data on a cache (210) according to Claim 3, wherein said step of  
15 predicting includes one or more of the following steps:

predicting said event time based on said data type;

observing one or more previous user behaviour patterns; or

20 predicting said event time following a trigger on another event.

6. The method (400) of preloading data on a cache (210) according to Claim 3 further characterised in that  
25 the step of predicting comprises predicting a preload time, by said at least one preload function (255, 265) based on said predicted data type.

7. The method (400) of preloading data on a cache  
30 (210) according to Claim 6, wherein said predicted preload time is based on one or more of the following parameters:

(i) An estimate of a cache re-load rate;

(ii) An availability of a communications network resource (155);

(iii) A previously achieved cache reload rate;

(iv) A cost parameter of one or more available  
5 communications network resources, for example a resource at a location and/or at a time;

8. The method (400) of preloading data on a cache (210) according to Claim 1 further characterised by the  
10 steps of:

determining (425) a current time; and

calculating a subsequent event or preload time therefrom.

15 9. The method (400) of preloading data on a cache (210) according to Claim 1, wherein said step of calculating a safety margin includes the step of:

predicting (410) an uncertainty of an event time, for example based on said data type and/or prevailing  
20 network conditions.

10. The method (400) of preloading data on a cache (210) according to Claim 1, wherein said safety margin is either set manually or is based on a monitoring of  
25 previous event occurrences.

11. The method (400) of preloading data on a cache (210) according to Claim 1, wherein said event includes one or more of the following:

30 (i) A diarised event for said user;  
(ii) A task to be performed by said user;  
(iii) A personal interest identified for said user;

(iv) A routine behaviour pattern identified for said user;

(v) A predictable behaviour pattern identified for said user; or

5 (vi) A foreseeable behaviour pattern identified for said user.

12. The method (400) of preloading data on a cache (210) according to Claim 1 further characterised by a  
10 step, prior to said step of preloading, of:

determining and implementing a timing margin (Tmmdg) (330) to allow for potential unavailability of said communications network (155) before commencing said step of preloading.

15

13. The method (400) of preloading data on a cache (210) according to Claim 12 further characterised by the steps of:

calculating a safety margin of time;  
20 determining whether a predicted timing of an event is within a time period of less than or equal to the current time minus said safety margin and/or said timing margin; and

commencing (465) said step of preloading in  
25 response to a positive determination.

14. The method (400) of preloading data on a cache (210) according to Claim 3, the method further characterised by an intermediate step of;

30 determining (455) whether said cache has capacity to store said data to be preloaded.

15. The method (400) of preloading data on a cache (210) according to Claim 14 further characterised by a step, prior to said step of preloading, of:

5 determining (435) a preferred maximum time (Tmpl) (350) before said predicted event time when said step of preloading can commence.

16. The method (400) of preloading data on a cache (210) according to Claim 1 further characterised by the  
10 step of:

adapting one or more timing parameters (330, 350) continuously or dynamically in response to a change in the communication network or user behaviour profile.

15 17. The method (400) of preloading data on a cache (210) according to Claim 16 further characterised by the steps of:

applying one or more threshold values to said one or more timing parameters (330, 350) for:

20 determining an acceptable cache hit rate,  
and/or

determining a preload success rate, and  
adapting said one or more timing parameters (330,  
350) in response to said determination(s).

25

18. The method (400) of preloading data on a cache (210) according to Claim 1 further characterised by the steps of:

30 grouping data types into categories based on, for example, one or more of the following: said data types, a priority of said data type, a predicted event time for said data to be preloaded; and

scheduling a preloading operation of data based on said grouping.

19. The method (400) of preloading data on a cache  
5 (210) according to Claim 1 further characterised by the step of:

determining (440) whether said cache has available capacity for receiving the preload data prior to commencing said step of preloading.

10

20. The method (400) of preloading data on a cache (210) according to Claim 19, wherein the step of determining whether said cache has available capacity includes measuring a rate of cache re-loads.

15

21. The method (400) of preloading data on a cache (210) according to Claim 8 further characterised by the step of:

20 determining (445) whether the current time is an economical time to preload said data to said cache, and in response to a positive determination, preloading said data to said cache (210).

22. The method (400) of preloading data on a cache  
25 (210) according to Claim 21, wherein the step of determining whether the current time is an economical time includes calculating whether a more economical time may be subsequently available within an acceptable preload window for said step of preloading.

30

23. The method (400) of preloading data on a cache (210) according to Claim 21, the method further characterised by the step of:

downloading one or more cost parameters associated with one or more network resource(s) to said host machine (240) or said local machine (235) or a remote server accessible by said host machine (240) or  
5 said local machine (235), such that said determination of whether said current time is an economical time to preload said data to said cache (210) can be made.

24. The method (400) of preloading data on a cache  
10 (210) according to Claim 1, wherein said step of preloading includes:

preloading said retrieved data in said cache (210), based on said user behaviour profile for said local machine (235), only when network costs are  
15 inexpensive, such that said data is made available to said cache user when desired at a substantially minimised cost.

25. The method (400) of preloading data on a cache  
20 (210) according to Claim 1 further characterised by the step of:

determining (450) whether a communications network (155) to be used in said preloading step is busy or whether said communications network (155) would be  
25 overloaded when commencing the preload operation, and in response to a positive determination delaying said step of preloading said cache (210).

26. The method (400) of preloading data on a cache  
30 (210) according to Claim 25, wherein, in response to determining that the communications network (155) is busy or would be overloaded, the method is further characterised by the steps of:

scheduling an entire preload operation for periods when the communication network is not busy; or  
scheduling said step of preloading on a block-by-block basis that provides intervals between said blocks  
5 for other users to use said communications network (155).

27. A cache (210) preloaded in accordance with Claim 1.

28. A local machine (235) characterised by a cache  
10 preload function (255) operably coupled to a cache (210) that is preloaded in accordance with Claim 1.

29. A local machine (235) comprising:  
a local communication unit (115) for operably  
15 coupling said local machine to a host machine (240) via a communication network (155); and

a cache (210) operably coupled to said local communication unit (115);  
the local machine (235) characterised by:  
20 a preload function (255), operably coupled to said cache (210), for determining a user behaviour profile for said local machine (235), predicting a time for data to be required by a user; calculating a safety margin of time retrieving data relating to said user  
25 behaviour profile from said data store (130) in response to said predicted time, and preloading data on said cache (210) based on said user behaviour profile, at a time at or before said safety margin prior to said predicted preload time, such that said data is made available to  
30 said cache user when desired.

30. The local machine (235) according to Claim 29, wherein said local machine (235) is a personal digital



assistant configured to communicate over, for example, a General packet radio network wireless network to a remote host machine (240).

5 31. A host machine (240) comprising:

a host communication unit (120) for operably coupling said host machine (240) to a local machine (235) via a communication network (155); and

10 a data store (130), operably coupled to said host communication unit (120);

the host machine (240) characterised by:

a preload function (265), operably coupled to said data store (130), for determining a user behaviour profile for said local machine (235), predicting a time  
15 for data to be required by a user, calculating a safety margin of time, retrieving data relating to said user behaviour profile from said data store (130) in response to a predicted time and preloading data from said data store (130) to a cache (210) on said local machine (235)  
20 based on said user behaviour profile, at a time at or before said safety margin prior to said predicted preload time, such that said data is made available to a user of said cache when desired.

25 32. A host machine (240) characterised by a data preload function (265) operably coupled to a data store (130), for performing the cache preload steps according to Claim 1.

30 33. A communications system (200) adapted to support the method (400) of preloading data on a cache (210) in a local machine (235) according to Claim 1.

34. A communications system (200) adapted to support  
a local machine (235) according to Claim 29.
35. A communications system (200) adapted to support  
5 a local machine (235) according to Claim 30.
36. A communications system (200) adapted to support  
a host machine (240) according to Claim 31
- 10 37. A communications system (200) adapted to support  
a host machine (240) according to or Claim 32.
38. A storage medium storing processor-implementable  
instructions for controlling a processor to carry out the  
15 method of Claim 1.